AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

List of Claims

1-30. (Canceled)

31. (Previously Presented) A processing system having a main memory partitioned into a plurality of partitions, each partition corresponding to a ranges of addresses in the main memory, the system comprising:

a plurality of processors;

a plurality of cache units;

a crossbar interface between the plurality of processors and the plurality of cache units, respectively;

a plurality of tables for the plurality of processors, respectively;

each table identifying

which of the plurality of cache units are available to the corresponding processor, and

which of the plurality of partitions are available to the identified cache units, respectively.

32. (Previously Presented) The system of claim 31, wherein the plurality of partitions represent non-overlapping ranges of main-memory addresses.

33. (Previously Presented) The system of claim 31, wherein:

the plurality of processors includes at least a first processor and a second processor;

the plurality of tables includes a first table for the first processor and a second table for the second processor; and

the first table and the second table express asymmetric mappings.

34. (Currently Amended) The system of claim 33, wherein:

the first table represents a first mapping to the plurality of cache units and a second mapping to the plurality of partitions; and

the second table represents a third mapping to the plurality of cache units and a fourth mapping to the plurality of partitions; and

at least one of the following is true, namely

the first mapping maps to different cache units than the ~~second~~ third mapping, and

the ~~third~~ second mapping maps to different partitions than the fourth mapping.

35. (Previously Presented) The system of claim 33, wherein:

the first and second tables identify in common at least one of the plurality of cache units; and

the second table identifies for the at-least-one common cache unit a different set among the plurality of partitions than is identified by the first table.

36. (Previously Presented) The system of claim 33, wherein:

the first and second tables identify in common at least one of the plurality of partitions;

the first table identifies a first set among the plurality of cache units for which the at-least-one common partition is available; and

the second table identifies a second set among the plurality of cache units for which the at-least-one common partition is available, the second set being different than the first set.

37. (Previously Presented) The system of claim 31, wherein

the table for at least one processor indicates that fewer than all of the plurality of cache units are available to the processor.

38. (Previously Presented) The system of claim 31, wherein

the table for at least one processor indicates that fewer than all of the plurality of partitions are mapped to the identified cache units.

39. (Previously Presented) The system of claim 31, wherein:

the plurality of processors, the plurality of cache units; the crossbar interface and the plurality of tables are configured on a single die.

40. (Previously Presented) The system of claim 31, wherein:

each of the plurality of tables is programmable to change at least one of the following, namely

which of the plurality of cache units are available to the corresponding processor, and

which of the plurality of partitions are available to the identified cache units, respectively.

41. (Previously Presented) A method mapping via a crossbar interface between a plurality of processors, a plurality of cache units and a main memory partitioned into a plurality of partitions, each partition corresponding to a ranges of addresses in the main memory, the method comprising:

providing a plurality of tables for the plurality of processors, respectively;

programming each table to identify which of the plurality of cache units are available to the corresponding processor, and

programming each table to identify which of the plurality of partitions are available to the cache units identified therein, respectively.

42. (Previously Presented) The method of claim 41, wherein the plurality of partitions represent non-overlapping ranges of main-memory addresses.

43. (Previously Presented) The method of claim 41, wherein:

the plurality of processors includes at least a first processor and a second processor;

the plurality of tables includes a first table for the first processor and a second table for the second processor; and

the method further comprises

configuring the first table and the second table to express asymmetric mappings.

44. (Currently Amended) The method of claim 43, wherein:

the first table represents a first mapping to the plurality of cache units and a second mapping to the plurality of partitions; and

the second table represents a third mapping to the plurality of cache units and a fourth mapping to the plurality of partitions; and

the method further comprises at least one of the following, namely

configuring the first mapping to map cache units differently than the ~~second~~ third mapping, and

configuring the ~~third~~ second mapping to map partitions differently than the fourth mapping.

45. (Previously Presented) The method of claim 43, further comprising:

arranging the first and second tables to identify in common at least one of the plurality of cache units;

arranging the first table to identify for the at-least-one common cache unit a first set among the plurality of partitions; and

arranging the second table to identify for the at-least-one common cache unit a different set among the plurality of partitions than the first set.

46. (Previously Presented) The method of claim 43, further comprising:

arranging the first and second tables to identify in common at least one of the plurality of partitions;

arranging the first table to identify a first set among the plurality of cache units for which the at-least-one common partition is available; and

arranging the second table to identify a second set among the plurality of cache units for which the at-least-one common partition is available, the second set being different than the first set.

47. (Previously Presented) The method of claim 41, wherein

arranging the table for at least one processor to indicate that fewer than all of the plurality of cache units are available to the processor.

48. (Previously Presented) The method of claim 41, wherein

arranging the table for at least one processor to indicate that fewer than all of the plurality of partitions are mapped to the identified cache units.

49. (Previously Presented) The method of claim 41, wherein:

programming at least one of the plurality of tables to change at least one of the following, namely

which of the plurality of cache units are available to the corresponding processor; and

which of the plurality of partitions are available to the identified cache units, respectively.

50. (Previously Presented) A processing system comprising:

storage partition means for representing ranges of addresses in a main memory as a plurality of partitions;

plural means for processing;

plural means for caching;

crossbar means for interfacing between the plural means for processing and the plural means for caching, respectively;

plural table means, corresponding to the plural means for processing, for respectively identifying

which of the plural means for caching are available to the plural means for processing, respectively, and

which of the plurality of partitions are available to the identified ones of the plural means for caching, respectively.